IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF APPEALS

Assignee's Docket No.: 7204

Group Art Unit: 2764

Serial No.: 09/020,699

Examiner: Chinor M. Lee

Filing Date: February 9, 1998

Title: Method and Apparatus for Determining the Validity of a Data Processing Transaction

CERTIFICATE OF MAILING

I CERTIFY THAT THIS DOCUMENT IS ADDRESSED TO THE COMMISSIONER OF PATENTS AND TRADEMARKS, WASHINGTON, DC 20231, AND WILL BE DEPOSITED WITH THE U.S. POSTAL SERVICE, FIRST CLASS POSTAGE PREPAID, ON

AUGUST 16 2000

GREGORY A. WELTE

APPEAL BRIEF

A Summary of Argument Begins on Page 5

The fee for this Brief may be billed to Deposit Account 140 - 225, NCR Corporation.

1. REAL PARTY IN INTEREST

NCR Corporation.

2. RELATED APPEALS AND INTERFERENCES

None.

3. STATUS OF CLAIMS

Claims 1, 2, and 4 - 19 are pending, rejected, and appealed. An amendment, attempting to add claim 20, was submitted with the notice of appeal. The appeal was taken from the third rejection. No final rejection has been made.

4. STATUS OF AMENDMENTS

An amendment, attempting to add claim 20, was submitted with

the notice of appeal. The appeal was taken from the third rejection. No final rejection has been made, so no amendment after final has been submitted.

An amendment is herewith submitted which corrects an error in claim 4.

5. SUMMARY OF INVENTION

The invention concerns a verification system for customers at a kiosk, such as an Automated Teller Machine. Figure 1 shows an ATM incorporating the invention.

A customer submits an ID card to the card reader 24 in Figure 2, through slot 12 in Figure 1. (Specification, page 3, lines 5 and 10.) The card contains encrypted data, such as a PIN (Personal Identification Number), the customer's birth date, and a telephone number. (Page 3, lines 15 - 20.)

The system reads the data from the card, and asks the customer to enter the PIN on keypad 16 in Figure 1. (Bottom of page 3, top of page 4.) The system then asks the user to enter two digits of the other data on the card, such as the third and first digits of the telephone number. (Page 4, lines 1 - 5.) That is, the system asks for two pieces of data: the PIN, and something else.

When the user attempts to use the ID card at a later time, the procedure is repeated. However, the system asks for a **different** data: the PIN, and a different "something else." For example, at

the later usage, the system may request the PIN and the birth date. (Page 4, first full paragraph.)

In addition, if the user makes a mistake in entering data, the invention asks for additional data from the card, and makes an estimate of whether the mistake was a guess by a thief, or an honest mistake by the card holder. (Page 4, bottom.)

6. ISSUES

Prior Art Rejections

Whether claims "1 - 2, 3 - 10 - 14, and 16 - 19" are obvious under 35 USC § 103, based on Suzuki. The precise claims in question are not clear: a typographical error appears in the Office Action on page 4, section 11, and appears in the quotation just given. However, since

(1) claims 1 - 19 were pending previously,

(2) claim 3 was cancelled, and

(3) claims 11 and 15 are rejected separately,

it is therefore presumed that the remaining claims are rejected as obvious, based on Suzuki. Those claims are 1, 2, 4 - 10, 12 - 14, and 16 - 19.

Whether claim 11 is obvious under 35 USC § 103, based on Suzuki and Granzow.

Whether claim 15 is obvious under 35 USC § 103, based on Suzuki and Chapin, Jr.

Section 112 Objections

Whether Applicant is required to label Figures 1 and 2 as "Prior Art," in view of the fact that the Specification states that the invention is contained in those two Figures, and thus such a labeling would admit that the invention is contained within the prior art.

Whether claim 2 is subject to objection on the grounds that its subject matter is not contained within the "Detailed Description of the Invention," in view of the facts that

(1) claim 2 was originally submitted with the application,

(2) MPEP § 2163.06(c) states that the claims as filed are part of the disclosure, and amendment can be made to copy the claims into the Detailed Description, and

(3) Applicant has offered to make such an amendment, but the Examiner has made no response.

Whether claim 17 is subject to objection, on the grounds that it recites making an "evaluation" on whether a keypad entry by a customer is a "guess" or not, in view of the facts that

(1) any person skilled in the art can make such an "evaluation," and

(2) the claim does not state that the "evaluation" must be correct, or even be used, but only that an evaluation is made.

7. GROUPING OF CLAIMS

Under the "Grouping of Claims Rule," only a single group exists, namely, "1 - 2, 3 - 10 - 14, and 16 - 19." Applicant interprets that listing (contained on page 4, section 11, of the Office Action) as referring to claims 1, 2, 4 - 10, 12 - 14, and 16 - 19, as explained above.

No claims in this group stand or fall together.

8. ARGUMENT

Summary of Argument

The first Office Action, paper number 6, stated that claim 17 would be allowable, if re-written. That was done.

The independent claims are 1, 6, 14, and 17. They were all rejected as obvious, based on Suzuki.

The Office Action (page 4, bottom) admits that two claim recitations are absent from Suzuki:

(1) requesting a second data item, in addition to a PIN, from a customer, and

(2) receiving the second data item.

That, by itself, is sufficient to preclude the 103-rejection. However, additional recitations are absent from Suzuki. Many

claims recite

(3) checking the second item for accuracy and

(4) using that check, plus a previous check, to determine the overall validity of the transaction.

If the "second data item" is absent from Suzuki, **which is admitted**, then these two steps (3) and (4) must also be missing. The reason ? They require the "second data item."

Therefore, **FOUR CLAIM RECITATIONS ARE ABSENT FROM SUZUKI.**

The 103-rejection cannot stand.

All dependent claims depend from these claims.

End of Summary

RESPONSE TO CLAIM REJECTIONS

Claims 1, 2, 4 - 10, 12 - 14, and 16 - 19

Claims 1, 2, 4 - 10, 12 - 14, and 16 - 19 were rejected as obvious, based on Suzuki. In this group of claims, claims 1, 6, 14, and 17 are independent claims.

As to these independent claims, the rejection (page 4, bottom) admits that Suzuki does not show (1) a request for a subset of a second field of security data, and (2) receiving the subset.

Claim 1

This admission necessarily entails the **added admission** that **other** claim elements are absent from Suzuki. For example, claim 1(d) recites "checking the subset . . . against [stored security data]." Claim 1(e) recites utilizing this checking step in making an overall determination. If the "subset" of claim 1(d) is never received, as admitted, then it can never be used, as in claim 1(d) and (e).

Therefore, the rejection's admission that two claim elements are missing from Suzuki necessarily leads to the conclusion that other claim recitations, such as claim 1(d) and 1(e), are missing also.

The rejection cannot stand. MPEP 2143.03 states:

> To establish prima facie obviousness . . . **all the claim limitations** must be taught or suggested by the prior art.

Claim 1(c) is admitted to be absent from Suzuki. That absence means that claim 1(d) and (e) are also absent.

Claim 6

The following recitation of claim 6 is admitted to be absent from Suzuki:

> a data processing unit for
> . . .
> (iii) controlling the communication means to request a **second entry** of data containing a specified subset of less than all digits of specified security data from the user via the data entry means.

That absence requires the absence of subsequent recitations of claim 6 from Suzuki, namely:

> (iv) checking the second entry of data against a subset of a second stored field of security data, and
>
> (v) determining the validity of the transaction based upon results of the checks made of the first and second entries of data against the first and second stored fields of security data, respectively.

Therefore, more than just the two admitted elements are absent from claim 6.

Claim 14

The rejection admits that part of claim 14(d) is absent from Suzuki. That absence requires a conclusion that claim 14(e) is also absent from Suzuki.

Claim 17

The rejection admits that part of claim 17(d) is absent from Suzuki. That absence requires a conclusion that claim 17(e) and (f) are also absent from Suzuki.

Conclusion

Therefore, at least two, and sometimes three or four, recitations of the independent claims in this group are absent from Suzuki. The obviousness rejection cannot stand.

Rationale for Rejection is Self-Defeating

The rationale for the rejection is that the addition of the elements missing from Suzuki "provides additional security . . . without incurring significant requirements for either additional hardware or software." (Paper number 8, top of page 5.) Several problems exist with this rationale.

Perhaps the main problem is that the rejection is self-defeating. In effect, it states that, if you modify Suzuki, you

obtain something which is very useful, but at no significant expense. However, that is evidence of patentability.

Rationale is Factually Incorrect

A second problem is that it is factually incorrect. The rejection states that additional software is not required to perform the missing steps. However, the missing steps are performed by a computer system. Therefore, additional software is clearly required.

The rejection apparently asserts that the additional software would not be "significant." In response, Applicant points to, Introduction to Compiler Construction, by Thomas W. Parsons, page 2, copy attached, which states:

> It has been estimated that the average programmer can produce 10 lines of debugged code in a working day.
>
> (The significant word is "debugged." We can all write huge amounts of code in a much shorter time, but the additional time required for testing and debugging reduces the overall figure drastically.)

If the rejection wishes to assert that no "significant" software is required, then, in view of this text passage, Applicant submits that the rejection must explain

(1) How much code is required to accomplish the missing steps, and

> (2) Why the length of time, as computed by this text passage, required to generate the code is not "significant."

Rationale not Found in Prior Art

A third problem is that the rationale is not found in the prior art. A teaching **within the prior art** must be shown which suggests combining the references. See MPEP § 2143.01.

Rationale is a Naked Conclusion

A fourth problem is that the assertion that "additional security" is obtained by the addition of the missing elements to Suzuki is a naked conclusion. No evidence has been given. Nor has a definition of "security" been given.

Rationale Modifies Suzuki

A fifth problem is that the addition of the elements to Suzuki acts as a modification of Suzuki. MPEP § 2143.01, last paragraph, states:

> If the proposed modification or combination of the prior art would **change the principle of operation** of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious.

Remaining Claims

The remaining claims are dependant. They depend from a parent claim which, as explained above, is not shown in Suzuki.

Response to Objections

Figures 1 and 2

Objection was registered to Figures 1 and 2, on the grounds that they must be labeled "Prior Art."

In brief, the objection mistakes **appearance** for **substance**. The **housing** shown in Figure 1 may appear within the prior art. However, labeling Figure 1 as "prior art" acts as an admission that the **machinery** within that housing is within the prior art. That is not so: the Specification states that the **machinery** includes the present invention. Applicants are not required to admit a falsehood.

Explaining this further, Applicant points out that the Specification, page 3, states that the ATM 10 of Figure 1 contains a "data processing unit 22" shown in Figure 2. That "data processing unit 22" performs many of the computation steps recited in the claims. (Page 3, line 5 et seq.)

Therefore, there is no doubt whatsoever that Figure 2 is not shown within the prior art. As to Figure 1, if Applicant designates Figure 1 as "Prior Art," then Applicant enables infringers to raise the argument that the "data processing unit 22"

and its functions are also prior art, because the Specification states that the ATM 10 of Figure 1 contains the data processing unit.

Applicant is not required to admit, directly or indirectly, that the invention is contained within the prior art.

Applicant offers to resolve this apparent problem by adding a block to Figure 1, within the ATM 10, which is labeled "present invention."

Claim 2

A previous objection was made to original claim 2, on the grounds that the subject matter of claim 2 was not set forth in the Detailed Description. That objection may be accurate.

Applicant responded by pointing out that, under section 112, the claims are **part** of the Specification. Applicant offered to amend the Detailed Description, to include the subject matter of claim 2. The Examiner did not respond to this offer, but merely repeated the objection.

The objection fails to comply with MPEP § 2163.06(c), which states:

> The claims as filed in the original specification are part of the disclosure and therefore, if an application as originally filed contains a claim disclosing material not disclosed in the remainder of the specification, the applicant may amend the

> specification to include the claimed subject matter.

As stated above, Applicant offered to make such an amendment, but the Examiner made no response. Under the MPEP, the resolution to this objection is an amendment to the Specification.

Claim 17

Objection was registered to claim 17, which states:

> evaluating whether lack of agreement results from a keying error, or from guessing . . .

The "lack of agreement" refers to discrepancies between (1) a code entered by a customer, and (2) the code which is expected, or stored in the ATM.

The objection states,

> . . . how does the system . . . determine the difference between a keying error and guessing ?

In response, Applicant points out that the Specification, page 5, beginning with line 16, illustrates a procedure (called a "hierarchy") to evaluate whether a customer is guessing. On page 6, lines 15 - 18, the Specification gives a specific formula to "evaluate" whether the customer is guessing. If guessing is inferred, then "The result is to initiate a full validation

procedure."

Independent of the Specification's procedure, Applicant points out that any person skilled-in-the art can make an "evaluation" of whether an entered password is the result of a guess. For example, if the actual password is "typewriter," a guess may be inferred if three characters of an entered password deviate from the actual password. Thus, if "xxxewriter" were entered, it would be evaluated to be a guess.

Applicant emphasizes that claim 17 **DOES NOT** state that the "evaluation" must be 100 percent correct. (That may be impossible.) Any person skilled-in-the-art would know, from reading the Specification, that the "evaluation" is a probabilistic concept. It may be wrong sometimes.

Applicant submits that the objection mis-reads the claim. The objection asks,

> . . . how does the system . . . **determine** the difference between a keying error and guessing ?"

The objection is based on a false notion. The claim does not state that the system **determines** whether an error, or guess, was made. The claim states that an "evaluation" is made.

Explaining this further, Applicant points out that the claim language does not recite making a **conclusive determination** of whether the customer was guessing. Claim 17 recites making an

"evaluation."

In addition to the "hierarchy" given above, the Specification gives other examples. The Specification, page 5, lines 12 - 15, states:

> A final request may occur after one previous failed attempt if the failed attempt is **so far from a correct entry that the user is likely to be guessing the correct entry.**

This gives one criterion for "evaluating" whether the user is guessing: a type of "distance" (as in "so far") between the actual entry and the correct entry is assessed. Distance is clearly measured in the number of errors.

As another example, the Specification, bottom of page 4, states that, if a mistake is received, the customer can recover from the mistake, by, in effect, correctly answering additional questions. This provides another approach to "evaluating" whether the customer is guessing. If the customer makes a mistake, ask the customer for more information (more digits from the telephone number, more digits from the birth date.)

-- If the additional digits are correct, the "evaluation" would be this: a mistake was made previously.

-- If the additional digits are wrong, the "evaluation" would be this: the customer was,

and probably is, guessing.

Of course, the "evaluation" may be wrong. The customer may be the correct customer, but drunk.

Therefore, the "evaluation" is simply a decision of whether guessing occurs or not. The "evaluation" may be wrong. Nothing in the claim states that the "evaluation" must be correct, or "determinative." The Specification gives clear examples of how to perform the "evaluation."

Grouping of Claims

The Grouping of Claims Rule only applies to the group of claims 1, 2, 4 - 10, 14, and 16 - 19.

Claim 2 recites displaying the first and second entries. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 4 recites specific characteristics of the two entries. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 5 states that a field is stored on an ID card. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 6 recites a method. Claim 1 recites an apparatus. Even if the apparatus performs the method, which is not admitted, the two claims are separately patentable. For example, a restriction requirement is commonly given on an apparatus, and the method is performs.

In addition, claim 6 recites three "means," and specific functions performed by each. The applied reference does not show the overall recitations of this claim, including these recitations, nor do the other claims in this group contain these recitations.

Claim 7 states that the communication means includes a visual display. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 8 recites a further request for data made by an apparatus, when an error is detected, and then examination of the data by the apparatus. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 9 states that the nature of the request of claim 8 is determined by the nature of the errors. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 10 recites a card reader. The applied reference does

not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 14 recites reading data from a card and, prior to asking for any other identity data, asking the party to enter that data. The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 16 recites "Method according to claim 14 and, wherein lack of agreement between an entered data and a data read from the card suspends the transaction." The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 17 recites (1) reading first and second data from a card, (2) asking the party to enter the first data, (3) comparing the two data and, if they agree, asking the party to enter the second data, (4) comparing the second data with the data read from the card and, if they disagree, suspending the transaction. The applied reference does not show the overall recitations of this claim, including these recitation, nor do the other claims in this group contain these recitations.

Claim 18 recites a second, later transaction, wherein the user is asked to enter a second subset of data. (For example, in the

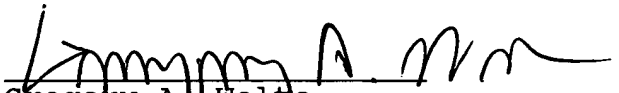
first transaction, the user may have been asked for the last two digits of a telephone number. In the later transaction, the user may be asked for the first two digits of that number.) The applied reference does not show the overall recitations of this claim, including this recitation, nor do the other claims in this group contain these recitations.

Claim 19 recites "the data processing unit, at one time, requests a specified subset A of digits of the security data, and, at another time, requests a specified subset B of digits of the same security data." The applied reference does not show the overall recitations of this claim, including these recitations, nor do the other claims in this group contain these recitations.

CONCLUSION

Applicant requests that the Board reverse the rejections and objections, and pass all claims to issue.

Respectfully submitted,



Gregory A. Welte
Reg. No. 30,434

NCR Corporation
101 West Schantz Avenue
ECD - 2
Dayton, OH 45479
August 16, 2000
(937) 445 - 4956

ATTACHMENT: Parsons

9. APPENDIX - Appealed Claims

1. A method of determining validity of a transaction carried out by a user at a data processing system, the method including the steps of:

a) receiving a user identification card and a first entry of data from the user;

b) checking the first entry of data against a first stored field of security data;

c) issuing a message to the user which requests a subset entry, consisting of less than all characters of a second stored field of security data, and receiving the subset entry from the user;

d) checking the subset entry against the corresponding subset within the second stored field of security data; and

e) determining the validity of the transaction based upon the results of the checks of steps (b) and (d).

2. A method according to claim 1, further comprising the step of:

f) displaying the first and second entries of data after receiving the second entry of data.

4. A method according to claim 3, wherein one entry of data is a personal identification number (PIN) associated with the user identification card and the other entry of data is data personal to an authorized holder of the card.

5. A method according to claim 4, wherein at least one of the first and second stored fields of security data is stored on the user identification card.

6. A data processing system for carrying out a transaction by a user of the system, the data processing system comprising:

manual data entry means for allowing the user to enter data;

communication means for communicating information to the user;

a data processing unit for

(i) controlling the communication means to request a first entry of data from the user via the data entry means,

(ii) checking the first entry of data against a first stored field of security data,

(iii) controlling the communication means to request a second entry of data containing a specified subset of less than all digits of

specified security data from the user via the data entry means,

(iv) checking the second entry of data against a subset of a second stored field of security data, and

(v) determining the validity of the transaction based upon results of the checks made of the first and second entries of data against the first and second stored fields of security data, respectively.

7. A data processing system according to claim 6, wherein the communication means includes visual display means for displaying the results of checking the first and second entries of data.

8. A data processing system according to claim 7, wherein the data processing unit causes the communication means for make at least one further request for data to be entered by the user through the data entry means when an incorrect entry of data is received, and then checks the data entered in response to the further request against stored security data.

9. A data processing system according to claim 8, wherein the nature of a further request for data is determined by the nature

of the error or errors in the data previously received from the user via the data entry means.

10. A data processing system according to claim 8, further comprising a card reader for reading data from a user identification card inserted by the user into the card reader for the purpose of initiating a transaction.

11. A data processing system according to claim 10, wherein the data processing unit causes the card reader to capture the user identification card when an error in the data is received in response to a final request.

12. A data processing system according to claim 11, wherein the card reader reads at least one of the stored fields of security data from the user identification card.

13. A data processing system according to claim 5, wherein the data processing unit keeps a record of the requested second entry of data.

14. A method of validating identity of a party attempting to execute a transaction, comprising the following steps:

a) accepting an identity card from the party;

b) reading first and second data from the card;

c) prior to asking for any other identity data, presenting a message asking the party to enter the first data; and

d) comparing the first data entered with the first data read from the card and, if they agree, presenting a message asking the party to enter the second data; and

e) comparing the second data entered with the second data read from the card and, if they agree, proceeding with the transaction.

15. Method according to claim 14, in which the first and second data are stored in the card in encrypted form.

16. Method according to claim 14 and, wherein lack of agreement between an entered data and a data read from the card suspends the transaction.

17. A method of validating identity of a party attempting to execute a transaction, comprising the following steps:

a) accepting an identity card from the party;

b) reading first and second encrypted data from the card;

c) presenting a message asking the party to enter the

first data; and

d) comparing the first data entered with the first data read from the card and, if they agree, presenting a message asking the party to enter the second data; and

e) comparing the second data entered with the second data read from the card and, if they agree, proceeding with the transaction

f) suspending the transaction if the second data entered fail to agree with data read from the card, and evaluating whether lack of agreement results from a keying error, or from guessing.

18. Method according to claim 1, wherein, at a later time, the user presents the identification card again, in connection with a different transaction, and method includes the step of

f) issuing a message requesting entry of a second subset entry, consisting of a different subset of said second stored field of security data.

19. System according to claim 6, wherein the data processing unit, at one time, requests a specified subset A of digits of the security data, and, at another time, requests a specified subset B of digits of the same security data.

The following claim was added in an amendment submitted with the notice of appeal.

20. Method according to claim 17, wherein the step of evaluating whether lack of agreement results from a keying error, or from guessing, comprises the step of

i) requesting further digits, and comparing the further digits with the second data read from the card.

INTRODUCTION TO COMPILER CONSTRUCTION

Thomas W. Parsons

Hofstra University

COMPUTER SCIENCE PRESS

An Imprint of W. H. Freeman and Company

New York

```
LR    3,5
```

instead of the bit string shown above. This representation of a program is known as *assembly language*. The assembler looked up the mnemonic LR in a table and found that the corresponding op-code was 00011000; it then found the binary representations of 3 and 5 and pieced the machine-language instruction together – that is, it *assembled* the instruction – from its component parts. In our example, the op-code concatenated with 0011 and 0101 gave the machine-language instruction shown above. A single line of assembly-language code normally corresponds to a single machine-language instruction.

Languages like Pascal, C, PL/I, FORTRAN, and COBOL are known, generally, as *high-level languages*: they have the property that a single statement, such as

```
x := y + z;
```

corresponds to more than one machine-language instruction. In particular, if x, y, and z are integers, and if the program is to be run on the IBM 370 or one of its congeners, then this statement will probably be translated into the sequence

```
L     3,Y         Load the working register with Y
A     3,Z         Add Z
ST    3,X         Store the result in X
```

or into the machine-language equivalent of these instructions.

The main virtues of high-level languages are productivity and the ability to manage complexity. It has been estimated that the average programmer can produce 10 lines of debugged code in a working day. (The significant word is "debugged": we can all write huge amounts of code in a much shorter time, but the additional time required for testing and debugging reduces the overall figure drastically.) It has also been found that this figure is essentially independent of the programming language used. Since a typical high-level language statement may correspond to perhaps 10 assembly-language instructions, it follows that we can be roughly 10 times as productive if we program in Pascal or C instead of assembly language.

Any program that has to deal with the real world will be complex. The real world teems with special cases, exceptions, and the general untidiness that distinguishes it from worlds like that of mathematics. Much of the evolution of programming languages has consisted of finding new ways of dealing with complexity that minimize its impact on the programmer. Such concepts as information hiding, encapsulation, and abstract data types are ways of coping with complexity. Many of these techniques can be implemented in assembly language, if you have the patience and the ingenuity, but they can be built into high-level languages in such a way that the implementation of the techniques is itself hidden from the user.